

CLAIMS

1. An object oriented computing environment comprising:

a first container based software component; said first container based
5 software component being an upgraded version of a second container based software component;

a container suitable for interaction with at least said first container based software component; and

10 an online upgrade system capable of operating to facilitate online upgrading of said second container based software component to said first container based software component.

2. A computing environment as recited in claim 1, wherein said online upgrade system comprises:

an online upgrade module;

an online upgrade control module capable of:

interacting with said online upgrade package to facilitate said online upgrading; and

15 interacting with an upgrade management entity to allow control or monitoring of said online upgrading operations by said upgrade management entity.

3. A computing environment as recited in claim 2,

25 wherein said control or monitoring of said online upgrading operations can be done at least partially by a human being.

4. A computing environment as recited in claim 3, wherein said human being is an application developer or a system administrator.

30 5. A computing environment as recited in claim 2, wherein said an online upgrade module comprises:

an online upgrade listener; and

an online upgrade specification.

6. A computing environment as recited in claim 5, wherein said first container based software component, said online upgrade listener; and said online upgrade specification are packaged together as a software package.
- 5 7. A computing environment as recited in claim 6,
wherein said online upgrade specification is implemented in a manifest portion of said software package.
8. A computing environment as recited in claim 7, wherein said first container based software component is an Enterprise Java Bean II application.
- 10 9. A computing environment as recited in claim 1, wherein said first container based software component can be installed without interrupting services that can be performed by said first or second container based software components.
10. A computing environment as recited in claim 1, wherein at least one of the first or second container based software components are operable while said first container based software component is being installed.
11. A computing environment as recited in claim 1, wherein said online upgrade system is capable of operating to facilitate online installation of said first container based software component in two or more stages.
- 20 12. A computing environment as recited in claim 11, wherein said two or more stages comprise:
an upgrade prepare stage;
a pre-upgrade stage;
an upgrade stage; and
a post-upgrade stage.
- 25 13. A computing environment as recited in claim 12, wherein said two or more stages further comprise of at least one a commit stage or a rollback stage.
- 30

14. A method of upgrading software in a object oriented computing environment;
said method comprising:

loading an online upgrade module; said online upgrade module including a
first container based software component, an online upgrade listener and an online
5 upgrade specification; said first container based software component being an
upgrade of a second container based software component which is capable of
operating in said object oriented computing environment;

notifying an online-upgrade controller to initiate an online upgrade process;

performing one or more operations to facilitate online upgrade of said second
10 container based software component to said first container based software
component.

15. A method as recited in claim 14, wherein said first container based software
component, said online upgrade listener; and said online upgrade specification
are packaged together as a software package.

16. A method as recited in claim 15, wherein said online upgrade specification is
implemented in a manifest portion of said software package.

17. A method as recited in claim 15, wherein said first container based software
component is an Enterprise Java Bean II software component.

18. A method as recited in claim 13, wherein said online upgrade can be performed
without interruption of services.

19. A method as recited in claim 13, wherein at least one of the first or second
container based software components are operable during said software upgrade.

20. A method of upgrading container based software components in multiple stages;
said method comprising:

an upgrade prepare stage;

a pre-upgrade stage;

one or more upgrade operations; and

a post-upgrade stage.

21. A method as recited in claim 20,
wherein said method further comprises a commit stage;
wherein said commit stage includes:

5 draining an older version of an application program;
 performing one or more callbacks; and
 unloading said older version of application program; and
 conveying information about said assuring, performing, or unloading to
a management entity.

10 22. A method as recited in claim 20, wherein said two or more stages further
comprise:
 a rollback stage.

15 23. A method as recited in claim 20, wherein said two or more stages further
comprise of at least one of a commit stage; and a rollback stage.

20 24. A method as recited in claim 20, wherein said upgrade prepare stage comprises:
 loading one or more listener classes;
 instantiating a listener; and
 performing at least one callback.

25 25. A method as recited in claim 21, wherein said upgrade prepare stage comprises:
 conveying information about said loading, instantiation, or performing to a
management entity.

 26. A method as recited in claim 25, wherein said management entity is a cluster
manager.

30 27. A method as recited in claim 20,
 wherein said upgrade pre-upgrade stage comprises performing one or more
callbacks;
 wherein said post-upgrade operations comprises one or more callbacks.

28. A method as recited in claim 27, wherein said method further comprises conveying information to a management entity.

29. A method as recited in claim 28, wherein said management entity is a cluster manager.

30. A method as recited in claim 20, wherein said method further comprises:

loading a first application program, said application program being a new version of a second application program;

performing a callback to determine whether the first application program is ready for service;

performing a redirect callback to the first application program; and

conveying information to a management entity about said loading, or said callbacks.